

AD-A039 586

PRITSKER AND ASSOCIATES INC WEST LAFAYETTE IND  
SAINT: A COMBINED DISCRETE/CONTINUOUS NETWORK SIMULATION TECHNI--ETC(U)  
APR 77 D J SEIFERT, D B WORTMAN, S D DUKE

F/G 5/8

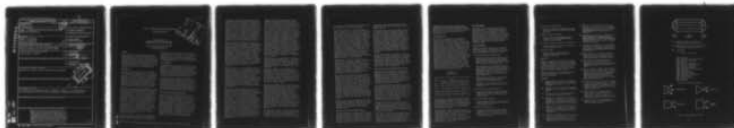
F33615-76-C-5012

UNCLASSIFIED

AMRL-TR-77-24

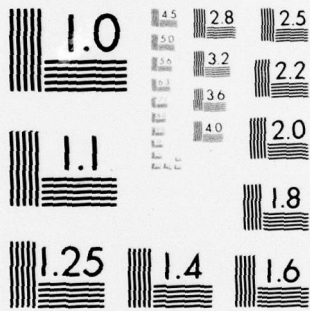
NL

1 of 1  
AD-A039 586



END

DATE  
FILMED  
6-77



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AMRL-TR-77-24	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SAINT: A COMBINED DISCRETE/CONTINUOUS NETWORK SIMULATION TECHNIQUE		5. TYPE OF REPORT & PERIOD COVERED Proceedings Reprint
7. AUTHOR(s) Deborah J. Seifert David B. Wortman** Steven D. Duket**		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Aerospace Medical Research Laboratory, Aerospace Medical Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433		8. CONTRACT OR GRANT NUMBER(s) In Part Under F33615-76-C-5012
11. CONTROLLING OFFICE NAME AND ADDRESS Same as block 9, above.		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62202F, 7184-13-03 2313-VI-21
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE April 1977
		13. NUMBER OF PAGES 6
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES ** Pritskers and Associates, Inc., Lafayette, Indiana. To be published in the Proceedings of the 8th Annual Conference on Modeling and Simulation.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) SAINT, Systems Analysis of Integrated Networks of Tasks, is a network modeling and simulation tech- nique designed to assist in the design and analysis of complex man-machine systems. SAINT provides the conceptual framework for representing systems that consist of discrete task elements, continuous state variables, and interactions between them. It facil- itates the assessment of the contribution system components make to overall system performance.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A U 39586

DDC FILE COPY

DDC  
APPROVED  
MAY 17 1977  
RECEIVED

391 382

679

SAINT: A COMBINED DISCRETE/CONTINUOUS NETWORK  
SIMULATION TECHNIQUE\*

Deborah J. Seifert  
Aerospace Medical Research Laboratory  
Wright-Patterson Air Force Base, Ohio

David B. Wortman and Steven D. Duket  
Pritsker & Associates, Inc.  
Lafayette, Indiana



ABSTRACT

SAINT, Systems Analysis of Integrated Networks of Tasks, is a network modeling and simulation technique designed to assist in the design and analysis of complex man-machine systems. SAINT provides the conceptual framework for representing systems that consist of discrete task elements, continuous state variables, and interactions between them. It facilitates the assessment of the contribution system components make to overall system performance.

INTRODUCTION

SAINT is a computer simulation tool for modeling and analyzing man-machine systems. SAINT consists of a symbol set for modeling systems and a computer program for analyzing such models. While SAINT was designed for modeling manned systems in which human performance is a major concern, it is potentially applicable to a broad class of problems--those in which discrete and continuous elements are to be portrayed and the behavior exhibits time varying properties. SAINT provides a mechanism for describing these dynamics so analyses can be performed.

SAINT evolved from two separate technologies. Task analysis and the Monte Carlo simulation of operator performance under workload stress as represented by Siegel and Wolf<sup>(1)</sup>, were the origin for the human factors development. Many of the features eventually incorporated into SAINT were identified as requirements based upon experience in applying this technology. The second origin of SAINT was in the GASP family of simulation techniques developed by Pritsker & Associates (e.g., GASP IV and P-GERTS, (2,3)). The earliest version of SAINT was an incorporation of the Siegel-Wolf model in a modified P-GERT package. The subsequent evolution of SAINT adapted features of GASP IV and allowed SAINT to become a flexible, sophisticated, combined modeling technique where networks of discrete events could be modeled along with the dynamics of continuous processes.

It is this ability to combine models of dynamics (e.g., aircraft equations of motion) with models of discrete activity sequences (e.g., operator actions) that permit the systems analyst to describe both

hardware and human performance in the context of a single model. This affords the system engineer the opportunity to analyze system effectiveness and isolate and hopefully quantify the relative contributions of man and machine. The SAINT concepts and terminology for performing such simulation analyses are described herein.

SAINT CONCEPTS

A graphical-network approach to modeling is taken, whereby a user of SAINT describes the system to be analyzed via a network model and auxiliary descriptions (e.g., equipment and operator performance parameters). The SAINT computer simulation program accepts a description of the network to be simulated and automatically performs an analysis to obtain statistical estimates of system performance.

A SAINT model contains two basic components. The discrete task-oriented component of the model is a network consisting of nodes, branches, resources, and information. The continuous state variable component is comprised of the mathematical descriptions of variables whose values change continuously over time. In addition, SAINT provides concepts for modeling interactions between the discrete task-oriented and continuous state variable components.

Discrete Task-Oriented Model Component

The discrete task-oriented component of the SAINT model is prepared in network form. The network consists of nodes and branches, each node representing a task. Tasks are described by a set of characteristics (e.g., performance time duration, priority, resource requirements). Branches connecting the nodes indicate precedence relations and are used to model the sequencing and looping requirements among the tasks. Complex precedence relations have been designed into SAINT to allow predecessor-successor relationships which are deterministic, probabilistic, and conditional. Resources, either human operators or hardware equipment, perform the tasks in accordance with the prescribed precedence relations, subject to resource availability.

\*This research has been supported by the United States Air Force under contract no. F33615-76-C-5012



In addition to specifying predecessor-successor requirements among tasks, the precedence relations indicate the flow of information through the network. Information is organized into packets with each information packet containing attributes that characterize the information being processed. The information packet can characterize items flowing through the network, signals being processed by the network, or any other concept related to network flow. When a task is completed, the information packet residing at the task is transmitted along each precedence branch selected. Information attribute values can be assigned or modified at any task in the network and can influence both task performance times and task branching relations.

Resources perform tasks either individually or in groups. Each resource included in a SAINT model is described by a set of attributes. These attributes are also organized into packets, with each packet characterizing a particular resource. Examples of operator attributes include such things as level of training, age, height, etc. Machine reliability is an example of an equipment attribute. Resource attributes are used in conjunction with the task descriptions in order to make a general network model resource-specific. The initial values of these user-defined resource attributes are assigned prior to the start of the simulation. The values may be dynamically changed at any task in the network and can be used as parameters in determining both task performance times and precedence relations.

In many instances it may be desirable to specify attributes which are not directly applicable to an information-oriented or resource-oriented characterization. These attributes are global in nature and do not flow through or move about the network as information and resource packets do. These attributes are defined as system attributes, with one set of system attributes being associated with a SAINT model. Temperature and time remaining in a mission are examples of model parameters which may be characterized as system attributes. Just as with information and resource attributes, system attributes may influence the task network performance and flow.

Each task in a SAINT network has two requirements which must be satisfied before the task can be performed. First, a specified number of predecessor tasks must be completed before the task is released. Second, once the task has been released, the resources required to perform the task must be available, (i.e., they are not busy performing other tasks). All tasks which have been released (all predecessor requirements have been satisfied) but whose required resources are not available are ranked in a queue according to their priority. Task priority may be assigned at the start of the simulation and may change dynamically as a function of system parameters and contingencies. When the required resources are made available due to task completions, the tasks in the waiting queue are started. The time to perform a task may be specified as a random variable defined by a probability distribution. SAINT supplies the user with eleven different distributions (Normal, Gamma, Beta, Weibull, etc.).

Frequently the task performance time is also a function of the type of task, the resource or resources performing the task, the status of the system, or the condition of the environment at the time the task is executed. SAINT provides for the specification of factors which influence task performance via user-written moderator functions. It is presumed the modeler can describe (e.g., by least squares techniques) the functional relationships between a set of conditions and a performance parameter or attribute of interest. For example, one might hypothesize that fatigue affects operator performance such that the average task time increases as a function of mission duration. The results of laboratory studies should provide a basis for postulating the form of this function and data appropriate for modeling the relationship. This empirically derived relationship can then be implemented in SAINT as a moderator function to determine the possible impact fatigue could have on operator performance, and in turn, on total system performance. In addition to moderator functions, user-written functions can be developed for specifying attribute assignments. Both types of functions are written in FORTRAN or a FORTRAN-compatible language.

SAINT provides for the development of dynamic and realistic system simulation models. Contingencies, decision making, and emergency conditions can be represented via SAINT's flexible attribute assignment and branching logic. SAINT provides two additional mechanisms for modeling dynamic system performance.

The first of these is termed task modification. This feature enables the user to modify task parameters as a function of ongoing system events. For example, consider a task which may require repetition due to a possibility of failure on the first attempt. The second time the task is performed the performance time may be significantly smaller than the initial execution. SAINT provides for the modification of the task time distribution after the initial attempt. Other task parameters can be modified in a similar fashion.

Clearing is the second SAINT modeling construct. Both tasks and resources can be cleared dynamically. "Task" clearing halts a specified task in progress, contingent on the completion of another task. "Resource" clearing halts whatever task the specified resource is performing. Both types of clearing may specify an additional task to be signaled. As an example, consider the simulation of an emergency condition in which all operators must stop their ongoing activities to assist in the emergency operations. This situation is best modeled in SAINT with resource clearing. The onset of the unexpected event would "free-up" (clear) the operators. Concurrently, emergency handling tasks would be initiated. Task and resource clearing provide dynamic realism in man-machine simulation modeling.

The symbol used to model a task in a SAINT network is illustrated in Figure 1. The input side of the node reflects the precedence requirements for releasing a task. The number of requirements for releasing a task the first time is on the top (PR1)

and the number of requirements for releasing a task on subsequent times is on the bottom (PR2).

The center portion of the task symbol shown in Figure 1 contains all task description information, such as performance time characteristics, statistics to be collected, and attributes to be assigned. It is subdivided into rows, with each row containing a specific type of descriptive information about the task. Further, each row is divided into two parts. The left-hand part contains the task description code. It is used to identify the type of information that appears in the right-hand part of the row, and can be any of the 17 available codes shown in Figure 2. By using the description code, only the information necessary to describe a task need be shown on the task symbol. In this manner, any or all of the task description codes can be specified for a particular task. If more than the four rows provided are required for a complete description, the user simply adds the necessary number of additional rows to the bottom of the task description portion of the task symbol.

The output side of the node contains the task number (TSK). In addition, the shape of the output side indicates the branching operation to be performed upon task completion. It specifies the process to be employed in selecting the successor tasks whose precedence requirements should be reduced by one. The four branching types included in SAINT are deterministic, probabilistic, conditional take-first, and conditional take-all. Their shapes are depicted in Figure 3.

When deterministic branching is specified, the number of requirements for all successor tasks is reduced by one. For probabilistic branching, each branch emanating from the task has an associated probability of selection. These probabilities may be specified directly or obtained from information, operator, or system attributes. Only a single successor task is selected. For conditional take-first branching, each branch has an associated condition, and the branches are ordered. Each condition is tested in the prescribed order, and the first branch whose condition is satisfied is selected. Conditional take-all branching operates in the same manner, but selects all branches whose conditions are satisfied. Conditions may be based on task completions, simulated time, or attribute values.

The above discussion only included the basic task node symbology. Additional symbolism is available for task modification, task signaling as a result of task or resource clearing, task signaling resulting from a threshold crossing, and state variable monitors.

#### Continuous State Variable Model Component

The second component of a SAINT model is the state variable description. While variables describing the task-oriented model component, such as resource busy/idle status, change values at discrete points in time, state variables change values continuously over time. The SAINT user defines these state variables by writing the algebraic, differential,

or difference equations that govern their time-dependent behavior. The use of state variables in SAINT is optional.

The SAINT user writes the state variable equations in a FORTRAN subroutine (subroutine STATE). State variables represented by algebraic or difference equations are defined in subroutine STATE as SS(.) variables. Those represented by differential equations are written in terms of DD(.) variables. SAINT employs a Runge-Kutta-England (RKE) numerical algorithm to integrate the equations of subroutine STATE written in terms of the DD(.) variables (2). The RKE algorithm obtains a solution to a set of simultaneous first order ordinary differential equations. Higher order differential equations can be modeled by placing the equations in canonical form. Subroutine STATE can be used to model state variables using a combination of DD(.) and SS(.) variables.

In SAINT, simulated time is advanced in accordance with the type of system being modeled. If no state variables are included, simulated time is advanced from one task completion to the next. When state variables are included in the model, time is also incremented in steps between scheduled task completions for the purpose of updating the values of the state variables. The step size is a function of user-specified accuracy requirements.

#### Discrete and Continuous Component Interactions

SAINT provides the capabilities of discrete-event simulation via the task-oriented network model component. With the incorporation of methods for modeling state variables, SAINT offers those capabilities most often associated with analog or continuous simulation. In addition, SAINT provides a number of concepts for modeling and simulating interactions between the discrete network and the continuous state variable components.

The interactions between tasks and state variables are initiated either by tasks being completed or by state variables crossing specified threshold values. Upon the completion of a task, state variables may be discretely regulated by increasing or decreasing their values. In addition, task completions can change the values of logical variables which can be used to alter state variable equation forms or the network structure. In this manner the discrete task-oriented component of the model affects the continuous state variable component.

Threshold crossings by state variables can signal or initiate tasks. Thus the values of state variables can influence task performance characteristics and precedence relations. Threshold crossings can also change the values of logical variables which, in turn, can be used to alter equations forms or change task precedence.

As an example of discrete and continuous component interactions, consider a system in which a pilot must keep the aircraft altitude within specified constraints. The pilot's inputs are modeled as discrete tasks and the aircraft dynamics as continuous state variables. When the altitude state crosses the allowable threshold value, the



corresponding discrete pilot correction task is signaled. The pilot makes the appropriate input and regulates the state variable(s) which determine altitude. Thus, through this component interaction, the aircraft altitude is brought back within acceptable limits.

#### STATISTICAL OUTPUT

Once the model has been built, the modeler can impose a data collection structure to obtain information about his description of the system as it is exercised. A variety of data can be obtained; these fall into three major categories. The first type of output is a statistical description of the execution of specific nodes or collections of nodes. Table I provides some insight into the variety of possible measures one can create using the built-in features of SAINT. Since users can create their own functions for updating attributes and for moderating network parameters, it has been necessary to allow the user to collect his own statistics on those parts of the model which cannot be predefined because the user creates them himself. SAINT supplies statistical subroutines for collecting data on user-supplied parts of the model. Tabular summaries of the computed descriptive statistics can then be generated to portray the results of a single iteration, a set of iterations, or a series of iterated runs showing the trends induced by some systematic variation of run conditions.

TABLE I  
STATISTICS CODES

What	When			
	Release	Start	Completion	Clearing
First	FIR REL	FIR STA	FIR COM	FIR CLE
All	ALL REL	ALL STA	ALL COM	ALL CLE
Between	BET REL	BET STA	BET COM	BET CLE
Number	NUM REL	NUM STA	NUM COM	NUM CLE

The second type of output is a graphic portrayal of the probability and cumulative density functions for a distributed variable. They provide a quick look at the shape of the data. An experienced user can store the actual values on an external device; later, the data can be fed to a plotting package for reproducible drawings.

The third type of output are time traces of the state variables. Up to ten variables can be plotted on the same graph with user specified scale factors and plotting symbols. Tabled values of the variables may also be obtained. Here again, an experienced programmer can save these data for more precise plots, but the tabulated plot provided by SAINT equips the user to quickly examine the results of his simulation run.

#### THE SAINT PROGRAM

Development of the SAINT simulation package has been completed and is fully documented (4,5,6,7). SAINT was developed in ANSI standard FORTRAN and, consequently, is machine-independent. The user, however, must supply his own system-specific random number generator. The task network data is punched on cards in free-form. SAINT includes an extensive input error-checking feature to assist users in debugging their models. For production runs, users can select a more efficient nonerror-checking version of SAINT.

#### SAINT APPLICATIONS

SAINT has been used to analyze a wide variety of man-machine systems. The following is a list of some of the organizations which have applied SAINT. They have developed SAINT models in order to:

- Aerospace Medical Research Laboratory
  - Evaluate alternative system configurations in a multioperator Remotely Piloted Vehicle/Drone Control Facility (RPV/DCF) in which operators monitor and control the flight of RPV's through the use of visual (CRT) displays (8).
  - Provide flight control performance predictions for the Digital Avionics Information System (DAIS) cockpit configuration in which dedicated instruments, displays, and subsystem status displays have been replaced with interactive multipurpose displays and multifunction keyboard switching (9).
  - Analyze design proposals in a B-52 strategic navigation system involving complex crew activities and task management (10).
  - Demonstrate a psychological model of the theoretical stages of a choice reaction time task (11).
- Air Force Human Resources Laboratory
  - Explore the feasibility of employing computer simulation for evaluating human effects on nuclear systems safety in a missile loading operation (12).
- Air Force Weapons Laboratory
  - Examine workload sharing and nuclear radiation effects on pilot performance in an air-to-air refueling mission (13).
- Inland Steel Corporation
  - Investigate the effect of higher degrees of automation, different capacities of process limiting operations, and alternative task allocations on the operator's idle times in a hot strip mill (14).

- Marshall Space Flight Center
  - Evaluate alternative task sequences and human factors considerations in space shuttle mission operations.
- School of Aerospace Medicine
  - Provide crew vulnerability/survivability estimates of B-52 flight missions subject to nuclear radiation exposure.
- U.S. Dept. of Commerce Office of Telecommunications
  - Analyze communication frequency utilization in a railroad switching yard.
- New Mexico State
  - Compare theoretical human performance predictions with empirically derived performance data (15).

#### CONCLUSION

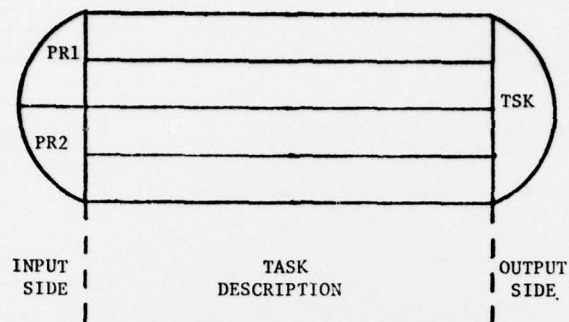
SAINT is a combined discrete/continuous network simulation technique based on GASP IV and P-GERT. It has the flexibility and complexity for modeling a surprising variety of applications at virtually any level of detail desired. These features provide the systems engineer with unique capabilities for modeling man-machine systems.

Additional information on the SAINT program and associated documentation can be obtained by contacting the authors.

#### REFERENCES

- (1) Siegel, Arthur K. and Wolf, J. Jay, Man-Machine Simulation Models: Performance and Psychosocial Interaction, John Wiley & Sons, Inc., 1967.
- (2) Pritsker, A. A. B., The GASP IV Simulation Language, John Wiley & Sons, Inc., New York, 1974.
- (3) Pritsker, A. A. B., P-GERT User's Manual, Pritsker & Associates, Inc., 1974.
- (4) Wortman, D. B., Duket, S. D., Hann, R. L., Chubb, G. P., and Seifert, D. J., Simulation Using SAINT: A User-Oriented Instruction Manual, Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, in preparation.
- (5) Wortman, D. B., Duket, S. D., Hann, R. L., Chubb, G. P., and Seifert, D. J., The SAINT User's Manual, Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, in preparation.
- (6) Wortman, D. B., Duket, S. D., Hann, R. L., Chubb, G. P., and Seifert, D. J., Documentation for the SAINT Simulation Program, Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, in preparation.
- (7) Wortman, D. B., Duket, S. D., Hann, R. L., Chubb, G. P., and Seifert, D. J., An Example to Illustrate the Use of SPSS for the Analysis of a SAINT Model, Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, in preparation.
- (8) Wortman, D. B., Duket, S. D., and Seifert, D. J., SAINT Simulation of a Remotely Piloted Vehicle/Drone Control Facility: Model Development and Analysis, Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, AMRL-TR-75-118, June 1976.
- (9) Kuperman, G. and Seifert, D. J., "Development of a Computer Simulation Model for Evaluating DAIS Concepts," Proceedings: Human Factors Society 19th Annual Meeting, Dallas, Oct 1975, pp. 347-353.
- (10) Chubb, Gerald P. and Berisford, Kathleen M., "Manned System Modeling: SAINT Applied to Strategic Navigation," Proceedings of the Tenth Annual Simulation Symposium, March 1977.
- (11) Hann, Reuben L. and Kuperman, Gilbert G., "SAINT Model of a Choice Reaction Time Paradigm," Proceedings: Human Factors Society 19th Annual Meeting, Dallas, Oct 1975, pp. 336-341.
- (12) Askren, William B., Campbell, Wendy Bohnett, Seifert, Deborah J., Hall, Thomas T., Johnson, Robert C., Sulzen, Robert H., Feasibility of a Computer Simulation Method for Evaluating Human Effects on Nuclear Systems Safety, Air Force Human Resources Laboratory, Air Force Systems Command, Brooks Air Force Base, AFHRL-TR-76-18, May 1976.
- (13) Wortman, David B., Sigal, C. Elliot, Pritsker, A. Alan B., and Seifert, Deborah J., New SAINT Concepts and the SAINT II Simulation Program, Section VI, Aerospace Medical Research Laboratory, Wright-Patterson AFB, Ohio, AMRL-TR-74-119, April 1975.
- (14) Maltas, Kenneth L. and Buck, James R., "Simulation of a Large Man/Machine Process Control System in the Steel Industry," Proceedings: Human Factors Society 19th Annual Meeting, Dallas, Oct 1975, pp. 193-205.
- (15) Teichner, Warren H., Human Performance Simulation, Annual Report, AFOSR Contract No. F44620-76-C-0013, Sept 1976.





PR 1 number of predecessor completions required for first release of the task

PR 2 number of predecessor completions required for subsequent release of the task

TSK task number

Figure 1. Task Symbol

LABL task label

TIME performance time characteristics

RESR resource requirements

PRTY priority

INCM information choice mode

DIFF different predecessor option

PREC completion precedence

ATAS attribute assignments

STCH system-specific characteristics

STAT statistics

MARK mark information

MODF moderation functions

DMOD distribution modification

TCLR task clearing

RCLR resource clearing

SWIT switching

REGL regulation

Figure 2. Task Description Codes

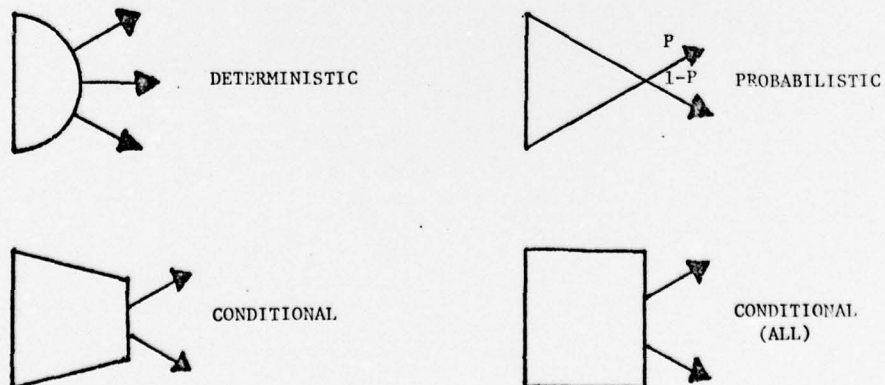


Figure 3. Task Branching Symbolism